

FIG. 1

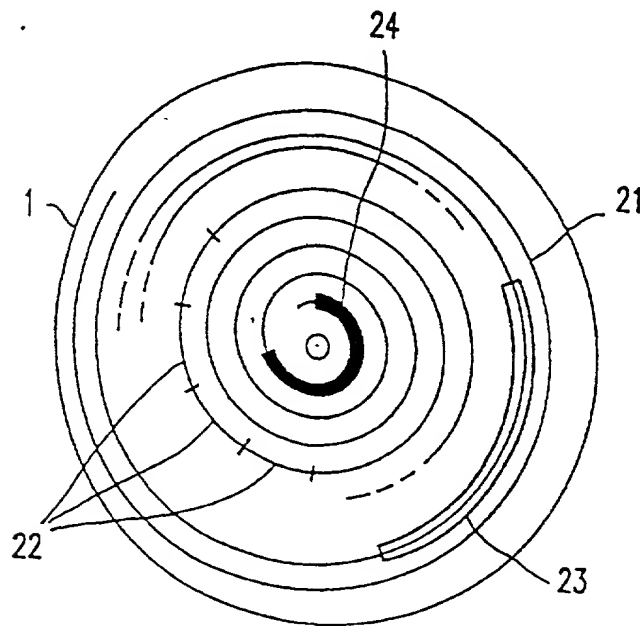


FIG. 2

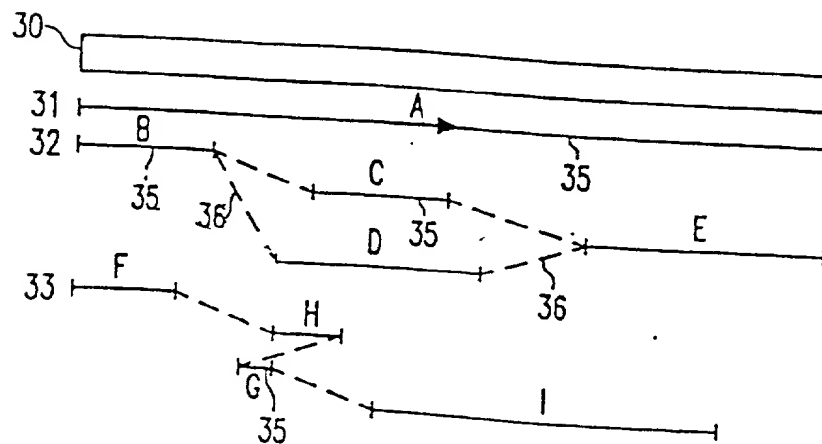


FIG. 3

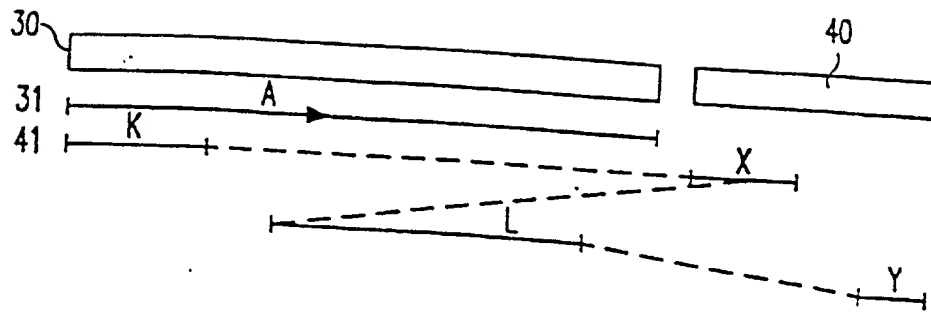


FIG. 4

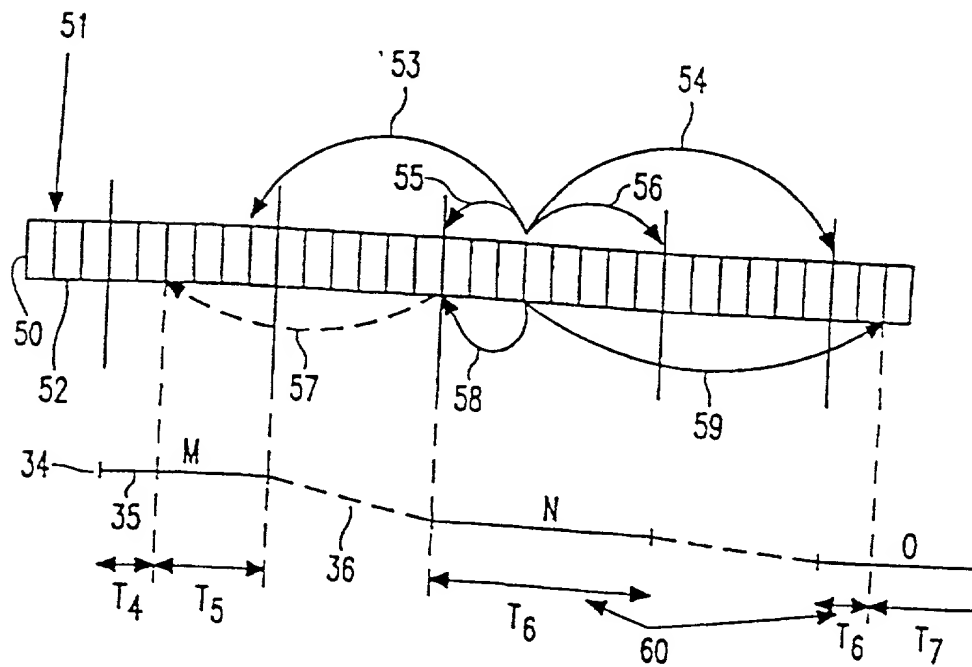


FIG. 5

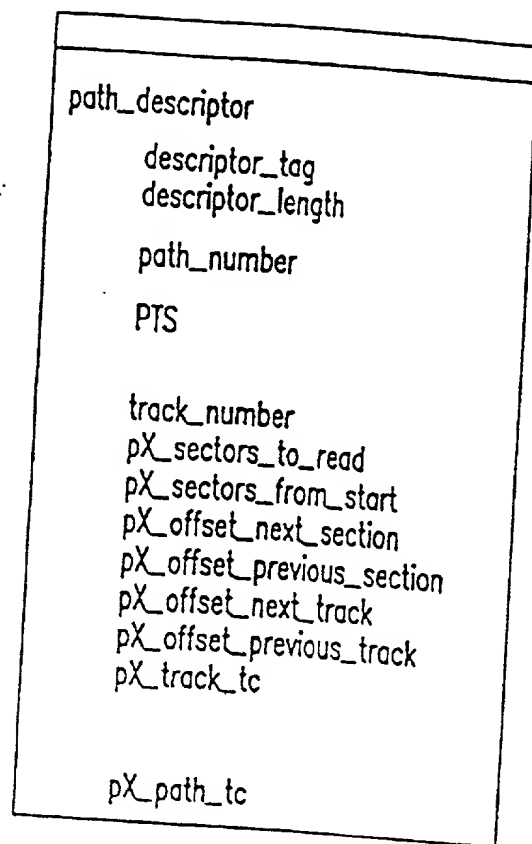


FIG. 6

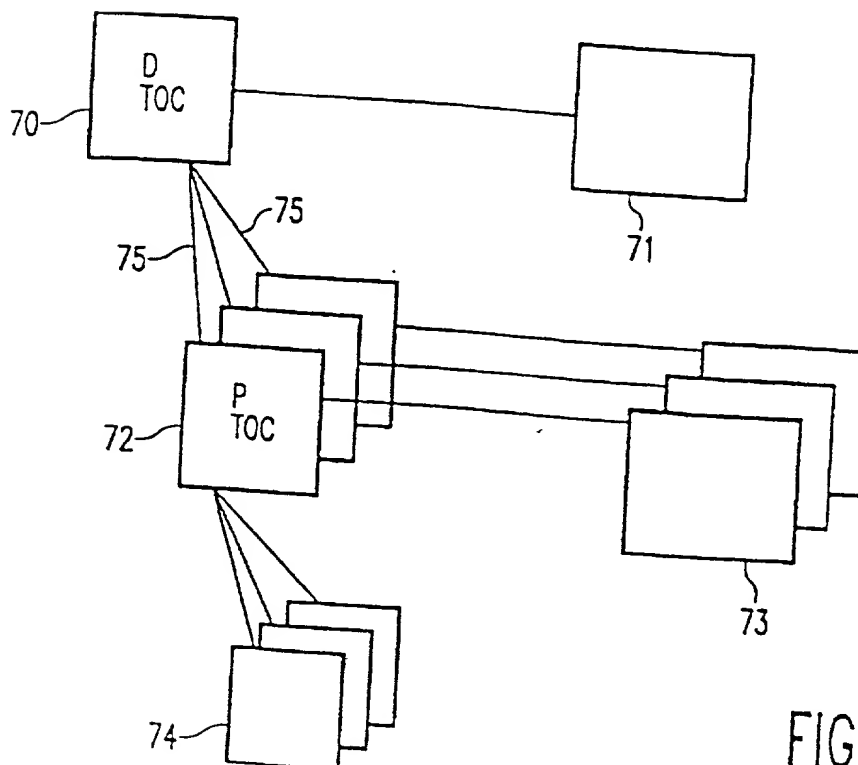
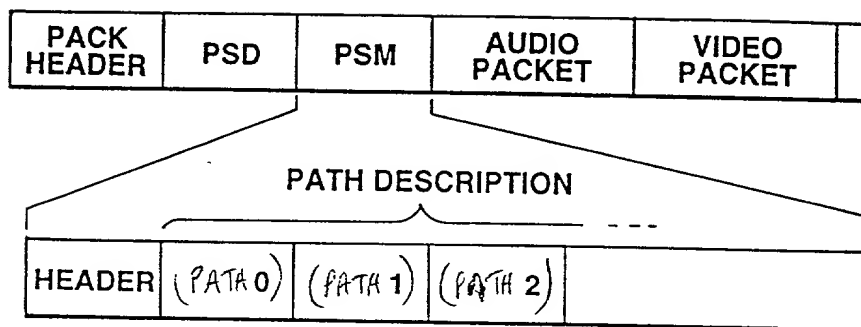


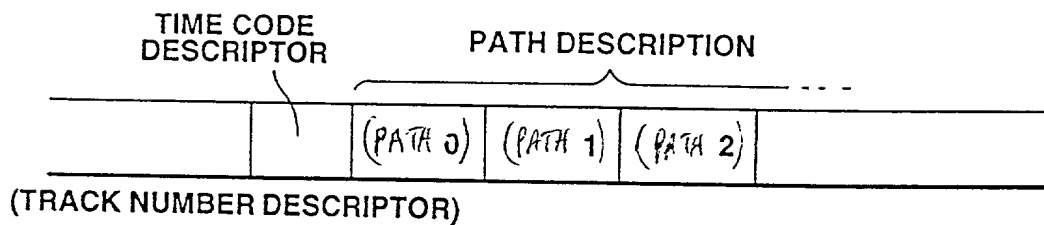
FIG. 7

Syntax	No. of bits	Mnemonic
path_descriptor()		
{		
descriptor_tag	8	uimbsf
descriptor_length	8	uimbsf
reserved	5	bslbf
path_number	3	uimbsf
reserved	4	bslbf
PTS[32..30]	3	bslbf
marker	1	bslbf
PTS[29..15]	15	bslbf
marker	1	bslbf
PTS[14..0]	15	bslbf
marker	1	bslbf
track_number	16	uimbsf
pX_sectors_to_read	24	uimbsf
pX_sectors_from_start	24	uimbsf
pX_offset_next_section	32	imsbf
pX_offset_previous_section	32	imsbf
pX_offset_next_track	32	imsbf
pX_offset_start_track	32	imsbf
pX_track_tc_hours_1	4	bslbf
pX_track_tc_hours_2	4	bslbf
pX_track_tc_minutes_1	4	bslbf
pX_track_tc_minutes_2	4	bslbf
pX_track_tc_seconds_1	4	bslbf
pX_track_tc_seconds_2	4	bslbf
pX_track_tc_CC	2	bslbf
pX_track_tc_frames_1	2	bslbf
pX_track_tc_frames_2	4	bslbf
pX_path_tc_hours_1	4	bslbf
pX_path_tc_hours_2	4	bslbf
pX_path_tc_minutes_1	4	bslbf
pX_path_tc_minutes_2	4	bslbf
pX_path_tc_seconds_1	4	bslbf
pX_path_tc_seconds_2	4	bslbf
pX_path_tc_CC	2	bslbf
pX_path_tc_frames_1	2	bslbf
pX_path_tc_frames_2	4	bslbf
}		

FIG. 6A



**FIG. 6B**



**FIG. 6C**

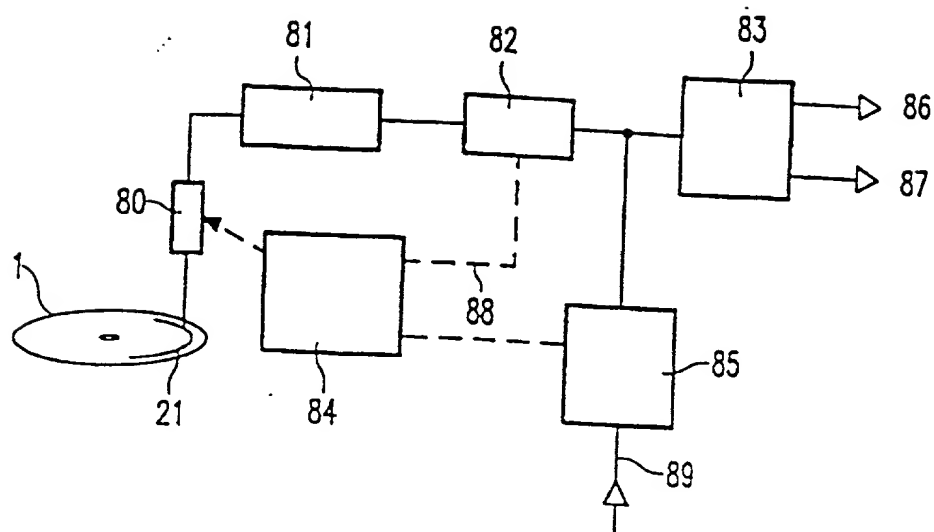


FIG. 8

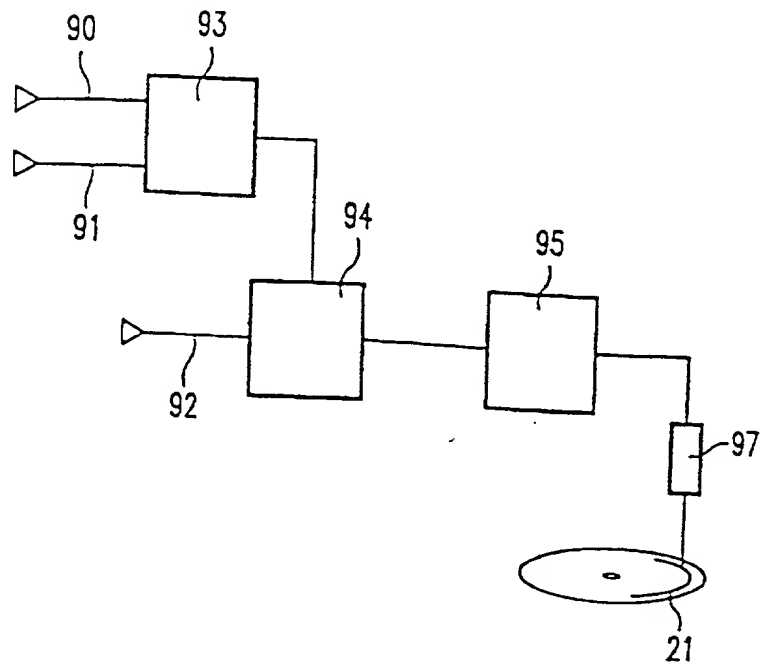


FIG. 9

FIG. 10

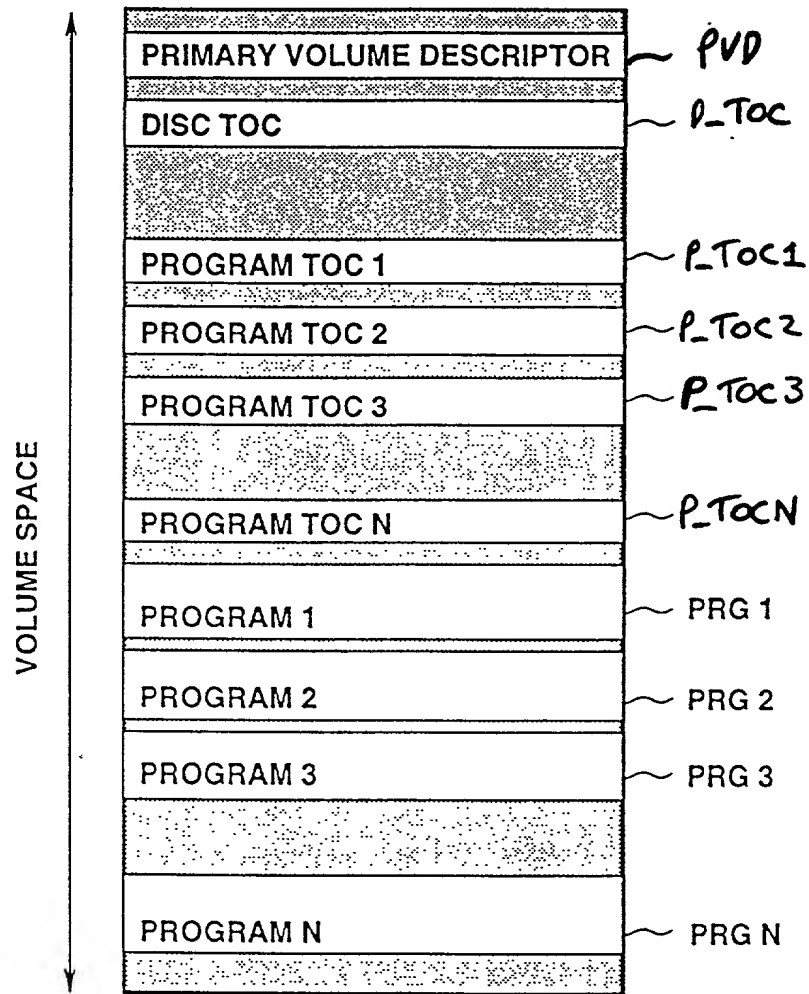


FIG.10

BP	Field name	Content
1	Volume Descriptor Type	numerical value
2 to 6	Standard Identifier	CD001
7	Volume Descriptor Version	numerical value
8	Unused Field	(00) byte
9 to 40	System Identifier	a-characters
41 to 72	Volume Identifier	d-characters
73 to 80	Unused Field	(00) bytes
81 to 88	Volume Space Size	numerical value
89 to 120	Unused Field	(00) bytes
121 to 124	Volume Set Size	numerical value
125 to 128	Volume Sequence Number	numerical value
129 to 132	Logical Block Size	numerical value
133 to 140	Path Table Size	numerical value
141 to 144	Location of Occurrence of Type L Path Table	numerical value
145 to 148	Location of Optional Occurrence of Type L Path Table	numerical value
149 to 152	Location of Occurrence of Type M Path Table	numerical value
153 to 156	Location of Optional Occurrence of Type M Path Table	numerical value
157 to 190	Directory Record for Root Directory	34 bytes
191 to 318	Volume Set Identifier	d-characters
319 to 446	Publisher Identifier	a-characters
447 to 574	Data Preparer Identifier	a-characters
575 to 702	Application Identifier	a-characters
703 to 739	Copyright File Identifier	d-characters, SEPARATOR 1, SEPARATOR 2
740 to 776	Abstract File Identifier	d-characters, SEPARATOR 1, SEPARATOR 2
777 to 813	Bibliographic File Identifier	d-characters, SEPARATOR 1, SEPARATOR 2
814 to 830	Volume Creation Date and Time	Digit(s), numerical value
831 to 847	Volume Modification Date and Time	Digit(s), numerical value
848 to 864	Volume Expiration Date and Time	Digit(s), numerical value
865 to 881	Volume Effective Date and Time	Digit(s), numerical value
882	File Structure Version	numerical value
883	(Reserved for future standardization)	(00) byte
884 to 1 395	Application Use	not specified
1 396 to 2 048	(Reserved for future standardization)	(00) bytes

FIG. 11



Syntax	No. of Bits	Mnemonic
disc_toc(){		
dvd_signature	64	bslbf
dvd_version	32	bslbf
length	16	uimsbf
toc_type	8	bslbf
reserved	8	bslbf
album()		
catalogue()		
reserved	16	uimsbf
number_of_programs (NOP)	16	bslbf
rating_definitions_offset	32	uimsbf
program_linkage_offset	32	uimsbf
disc_track_offset	32	uimsbf
program_toc_pointers_offset	32	uimsbf
disc_play_time_offset	32	uimsbf
disc_name_offset	32	uimsbf
disc_date_offset	32	uimsbf
disc_copyright_offset	32	uimsbf
disc_publisher_offset	32	uimsbf
reserved	288	bslbf
rating_definitions()		
program_linkage()		
disc_tracks()		
program_toc_pointers()		
disc_play_time()		
disc_name()		
disc_date()		
disc_copyright()		
disc_publisher()		

FIG. 12

Syntax	No. of Bits	Mnemonic
disc_tracks() {		
for(i=0; i < 8; i++){		
number_of_tracks(NOT)	16	uimsbf
reserved	16	bslbf
for (t=0; t < NOT; t++) {		
program_number	16	uimsbf
track_number	16	uimsbf
start_lsa	32	uimsbf
last_lsa	32	uimsbf
last_es_lsa	32	uimsbf
stoppiing_stc	32	uimsbf
playing_time()		
}		
}		
}		

FIG. 12A

Syntax	No. of Bits	Mnemonic
program_toc(){		
dvd_signature	64	bslbf
dvd_version	32	bslbf
length	16	uimsbf
toc_type	8	bslbf
reserved	8	bslbf
number_of_tracks (NOT)	16	uimsbf
reserved	16	bslbf
program_start_lsa	32	uimsbf
path_rating_assignment_offset	32	uimsbf
for (i=0; i<8; i++){		
program_tracks_offset_i	32	uimsbf
}		
elementary_stream_info_offset	32	uimsbf
program_play_times_offset	32	uimsbf
program_name_offset	32	uimsbf
track_names_offset	32	uimsbf
program_date_offset	32	uimsbf
track_dates_offset	32	uimsbf
program_copyright_offset	32	uimsbf
program_publisher_offset	32	uimsbf
reserved	288	bslbf
path_rating_assignments()		
for (i=0; i<8; i++){		
if (program_chapters_offset_i>0){		
program_tracks()		
entry_points()		
path_table()		
}		
}		
elementary_stream_info()		
program_play_times()		
program_name()		
track_names()		
program_date()		
track_dates()		
program_copyright()		
program_publisher()		

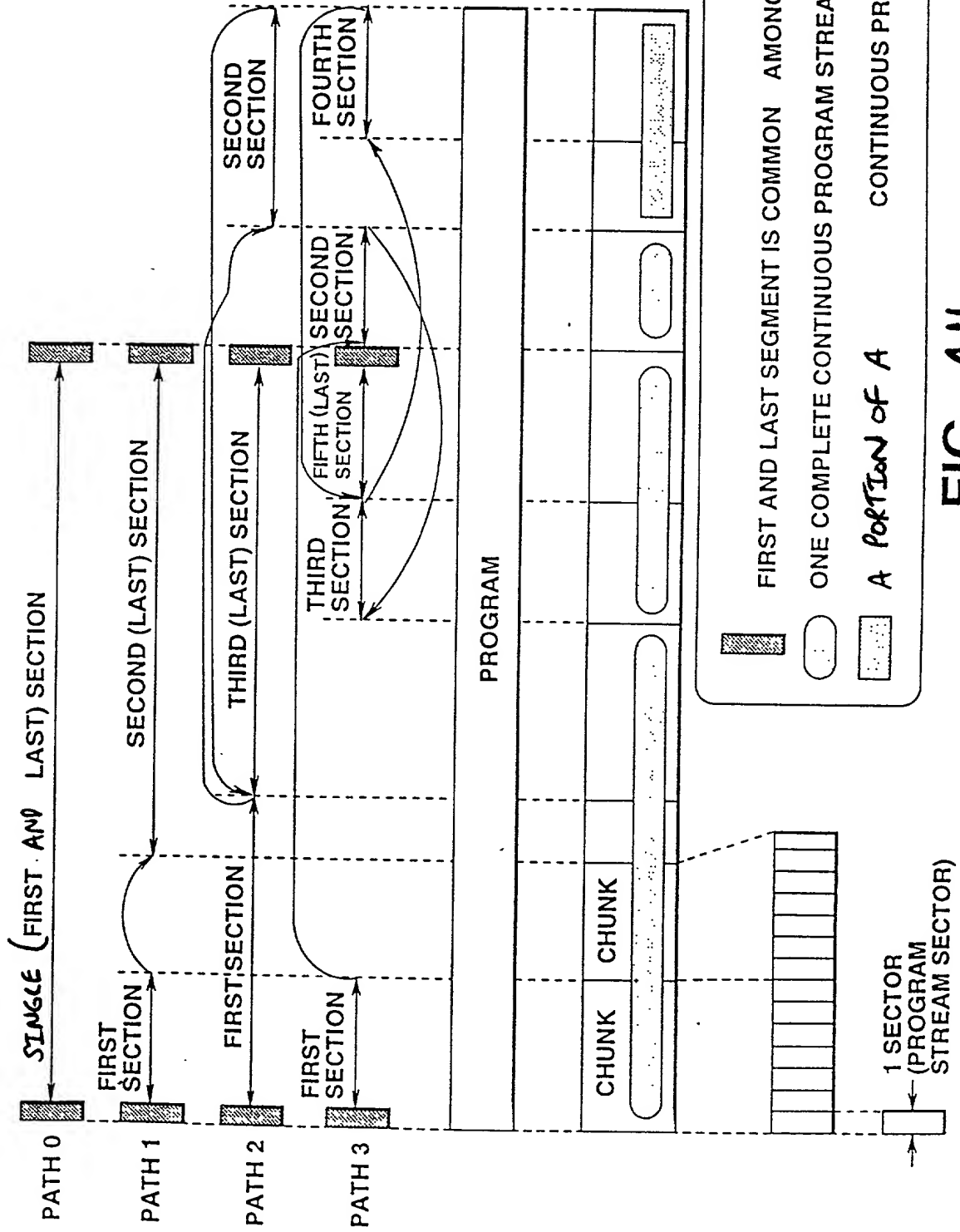
FIG. 13

Syntax	No. of Bits	Mnemonic
program_tracks() {		
number_of_path_tracks(NOPT)	16	uimsbf
reserved	16	bslbf
for (t=0; t < NOPT; t++) {		
reserved	16	bslbf
track_number	16	uimsbf
start_rsa	32	uimsbf
last_rsa	32	uimsbf
last_es_rsa	32	uimsbf
stoppi ng_stc	32	uimsbf
track_playing_time()	32	uimsbf
}		
}		

FIG. 13A

Syntax	No. of Bits	Mnemonic
path_table() {		
number_of_sections(NOCH)	32	uimsbf
for (ch=0; ch < NOCH; ch++) {		
start_rsa	32	uimsbf
last_rsa	32	uimsbf
stoppi ng_stc	32	uimsbf
last_es_rsa	32	uimsbf
}		
}		

FIG. 13B



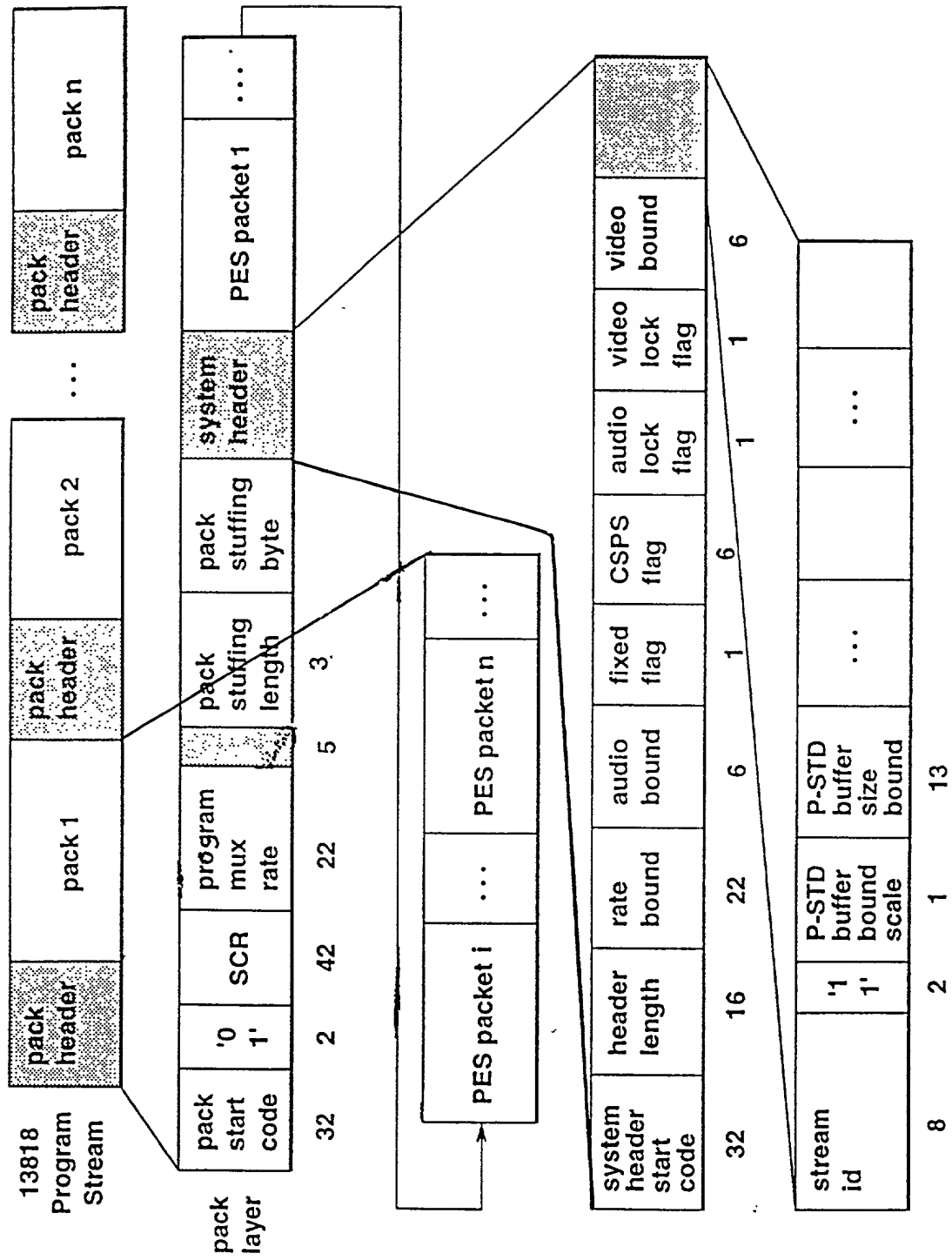


FIG. 15A

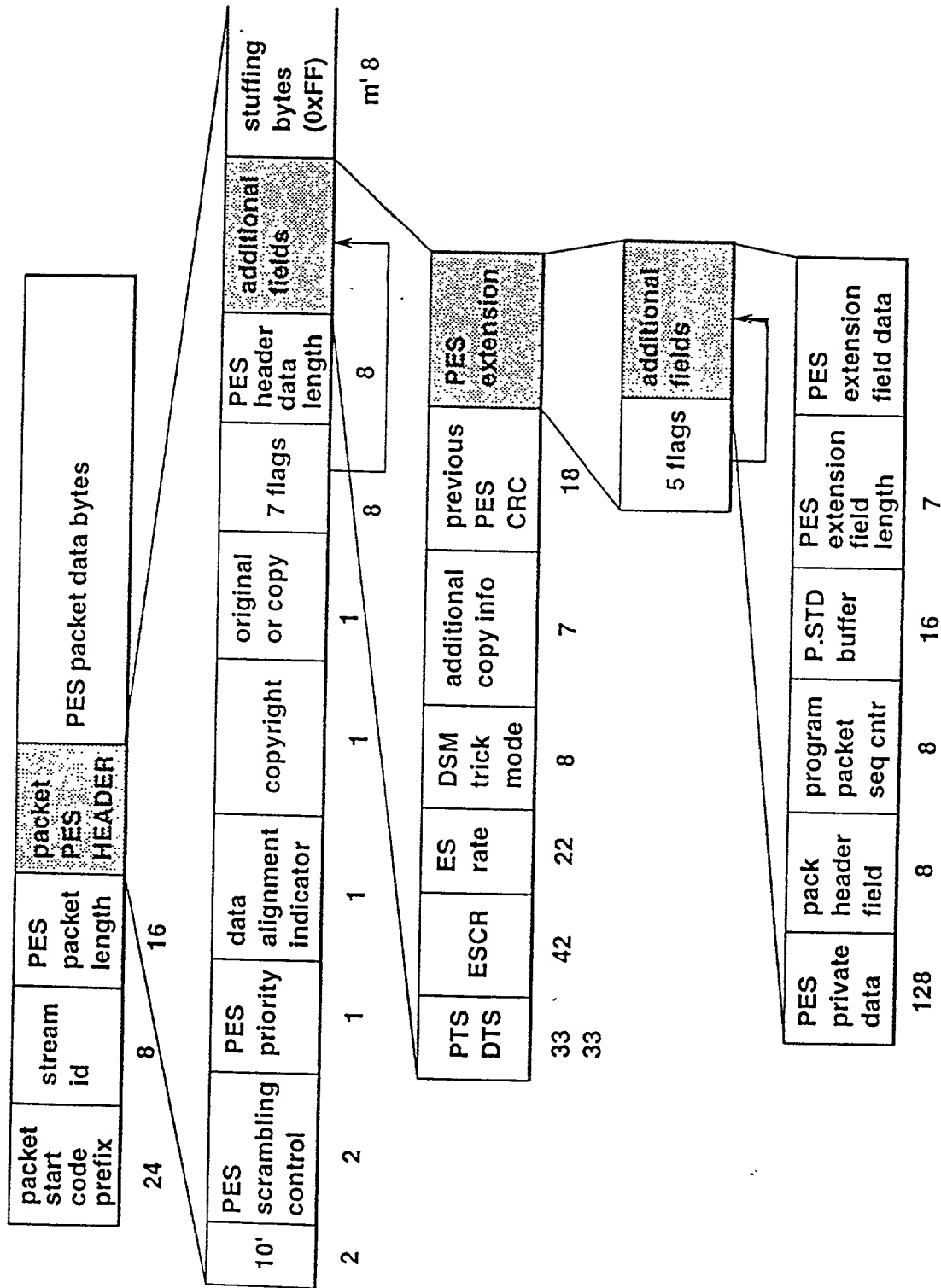


FIG. 15B

Syntax	No. of Bits	Mnemonic
PES_packet(){		
packet_start_code_prefix	24	bslbf
stream_id	8	uimsbf
PES_packet_length	16	uimsbf
if (stream_id != program_stream_map		
&& stream_id !=padding_stream		
&& stream_id !=private_stream_2		
&& stream_id !=ECM		
&& stream_id !=EMM		
&& stream_id !=program_stream_directory		
&& stream_id !=DSMCC_stream		
&& stream_id !=ITU-T Rec. H.222.1 type E_stream){		
'10'	2	bslbf
PES_scrambling_control	2	bslbf
PES_priority	1	bslbf
data_alignment_indicator	1	bslbf
copyright	1	bslbf
original_or_copy	1	bslbf
PTS_DTS_flags	2	bslbf
ESCR_flag	1	bslbf
ES_rate_flag	1	bslbf
DSM_trick_mode_flag	1	bslbf
additional_copy_info_flag	1	bslbf
PES_CRC_flag	1	bslbf
PES_extension_flag	1	bslbf
PES_header_data_length	8	uimsbf
if (PTS_DTS_flag == '10') {		
'0010'	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
if (PTS_DTS_flags == '11') {		
'0011'	4	bslbf
PTS [32..30]	3	bslbf
marker_bit	1	bslbf
PTS [29..15]	15	bslbf
marker_bit	1	bslbf
PTS [14..0]	15	bslbf
marker_bit	1	bslbf
'0001'	4	bslbf
DTS [32..30]	3	bslbf
marker_bit	1	bslbf
DTS [29..15]	15	bslbf

FIG. 16A



Syntax	No. of Bits	Mnemonic
marker_bit	1	bslbf
DTS [14..0]	15	bslbf
marker_bit	1	bslbf
}		
if (ESCR_flag == '1') {		
reserved	2	bslbf
ESCR_base [32..30]	3	bslbf
marker_bit	1	bslbf
ESCR_base [29..15]	15	bslbf
marker_bit	1	bslbf
ESCR_base [14..0]	15	bslbf
marker_bit	1	bslbf
ESCR_extension	9	uimsbf
marker_bit	1	bslbf
}		
if (ES_rate_flag == '1') {		
marker_bit	1	bslbf
ES_rate	22	uimsbf
marker_bit	1	bslbf
}		
if (DSM_trick_mode_flag == '1') {		
trick_mode_control	3	uimsbf
if (trick_mode_control == fast_forward) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
}		
else if (trick_mode_control == slow_motion) {		
rep_cntrl	5	uimsbf
}		
else if (trick_mode_control == freeze_frame) {		
field_id	2	uimsbf
reserved	3	bslbf
}		
else if (trick_mode_control == fast_reverse) {		
field_id	2	bslbf
intra_slice_refresh	1	bslbf
frequency_truncation	2	bslbf
else if (trick_mode_control == slow_reverse) {		
rep_cntrl	5	uimsbf
}		
else		
reserved	5	bslbf
}		
if (additional_copy_info_flag == '1') {		
marker_bit	1	bslbf
additional_copy_info	7	bslbf

FIG. 16B

Syntax	No. of Bits	Mnemonic
<pre> } if (PES_CRC_flag == '1') {     previous_PES_packet_CRC } if (PES_extension_flag == '1') {     PES_private_data_flag     pack_header_field_flag     program_packet_sequence_counter_flag     P-STD_buffer_flag     reserved     PES_extension_flag_2     if (PES_private_data_flag == '1') {         PES_private_data     }     if (pack_header_field_flag == '1') {         pack_field_length         pack_header()     }     if (program_packet_sequence_counter_flag == '1') {         marker_bit         program_packet_sequence_counter         marker_bit         MPEG1_MPEG2_identifier         original_stuff_length     }     if (P-STD_buffer_flag == '1') {         '01'         P-STD_buffer_scale         P-STD_buffer_size     }     if (PES_extension_flag_2 == '1') {         marker_bit         PES_extension_field_length         for (i=0; i&lt;PES_extension_field_length; i++)         {             reserved         }     }     for (i=0; i&lt;N1; i++) {         stuffing_byte     }     for (i=0; i&lt;N2; i++) {         PES_packet_data_byte </pre>	<pre> 16 1 1 1 1 3 1 128 8 1 7 1 1 6 2 1 13 1 7 8 8 8 </pre>	<pre> bslbf bslbf bslbf bslbf bslbf bslbf bslbf bslbf uimbsf bslbf uimbsf bslbf bslbf uimbsf bslbf bslbf uimbsf bslbf uimbsf bslbf bslbf bslbf </pre>

FIG.16 C

Syntax	No. of Bits	Mnemonic
<pre>         }     }     else if (stream_id == program_stream_map        stream_id == private_stream_2        stream_id == ECM        stream_id == EMM        stream_id == program_stream_directory        stream_id == DSMCC_stream)        stream_id == ITU-T Rec. H.222.1 type E stream {         for (i=0; i&lt;PES_packet_length; i++) {             PES_packet_data_byte         }     }     else if (stream_id == padding_stream) {         for (i=0; i&lt;PES_packet_length; i++) {             padding_byte         }     } } </pre>	<p>8</p> <p>8</p>	<p>bslbf</p> <p>bslbf</p>

FIG. 16 D

ONE MULTIPLEXED BITSTREAM

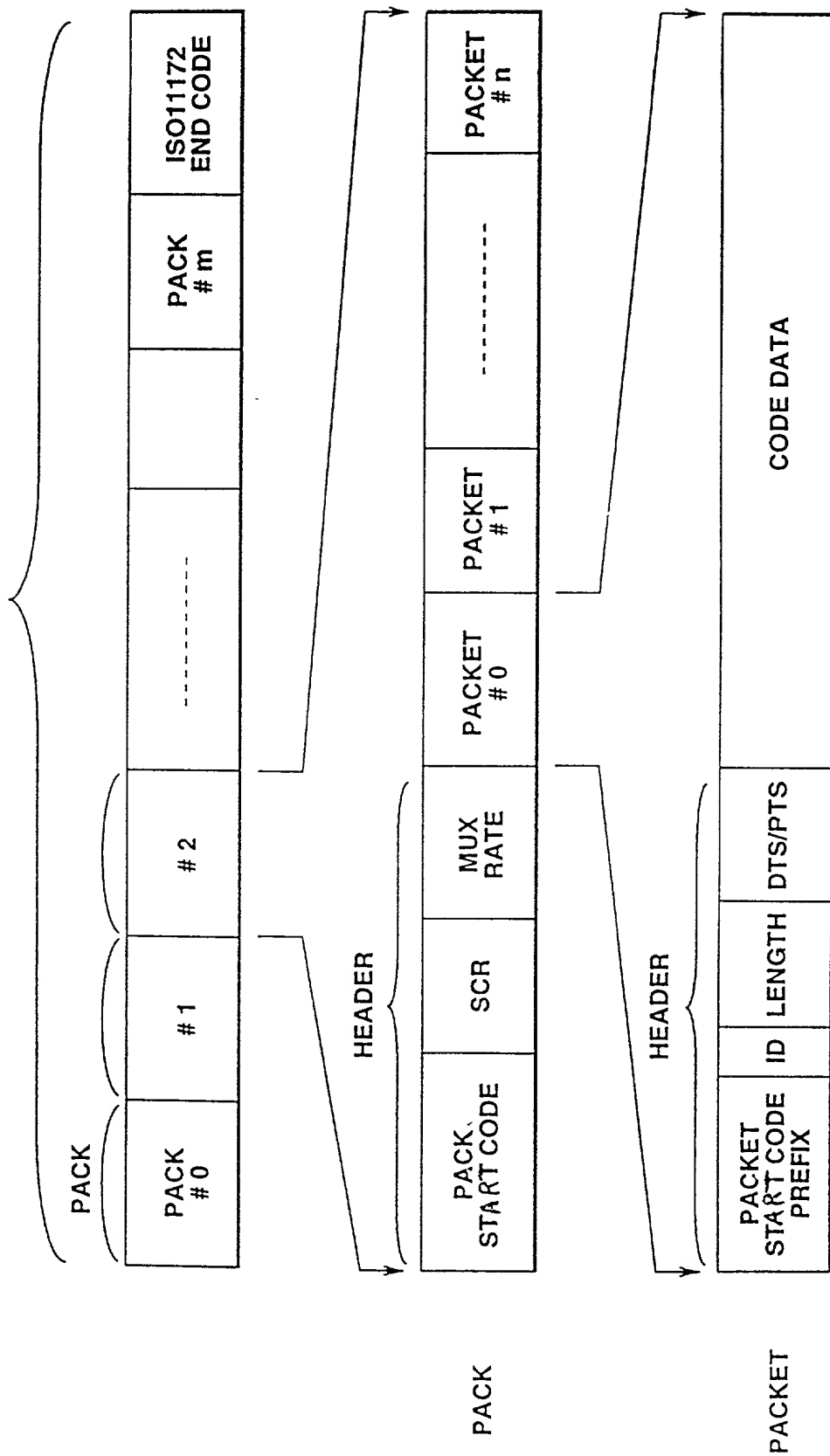


FIG.17

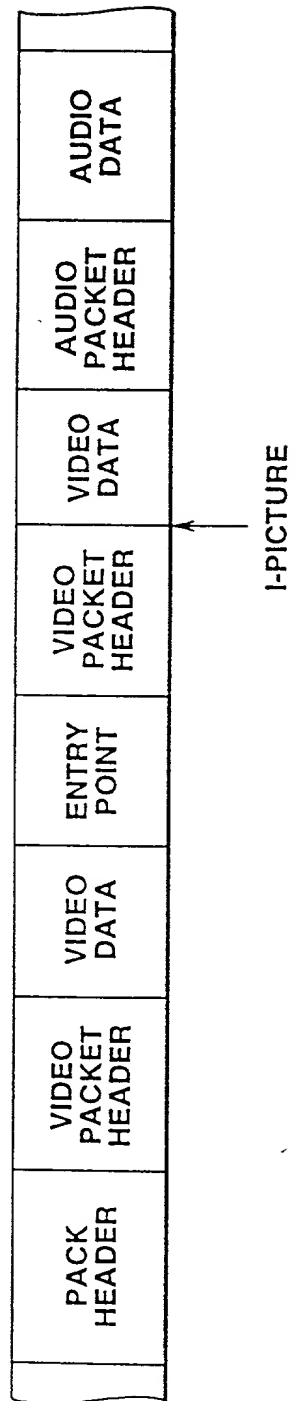


FIG. 18

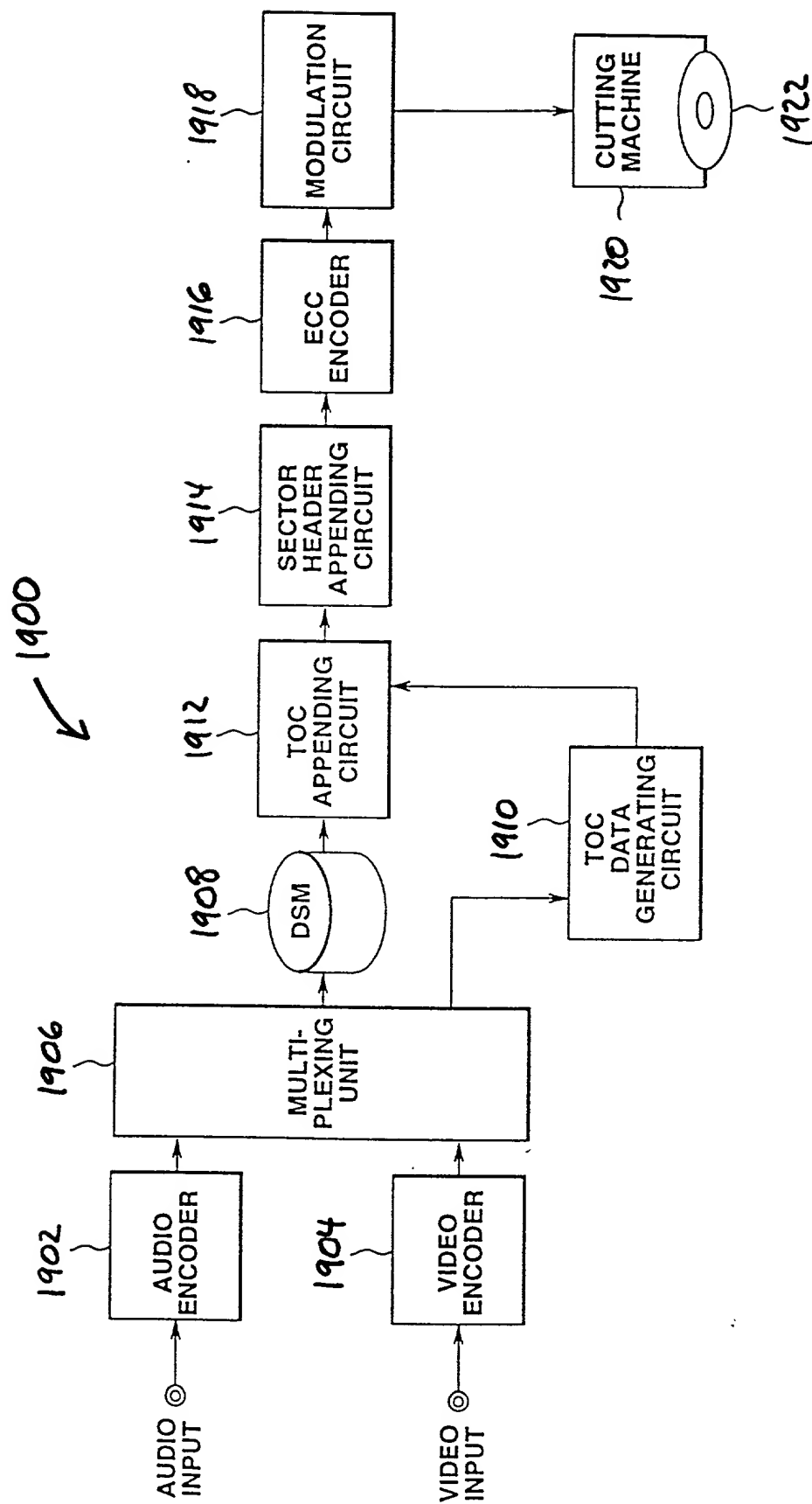
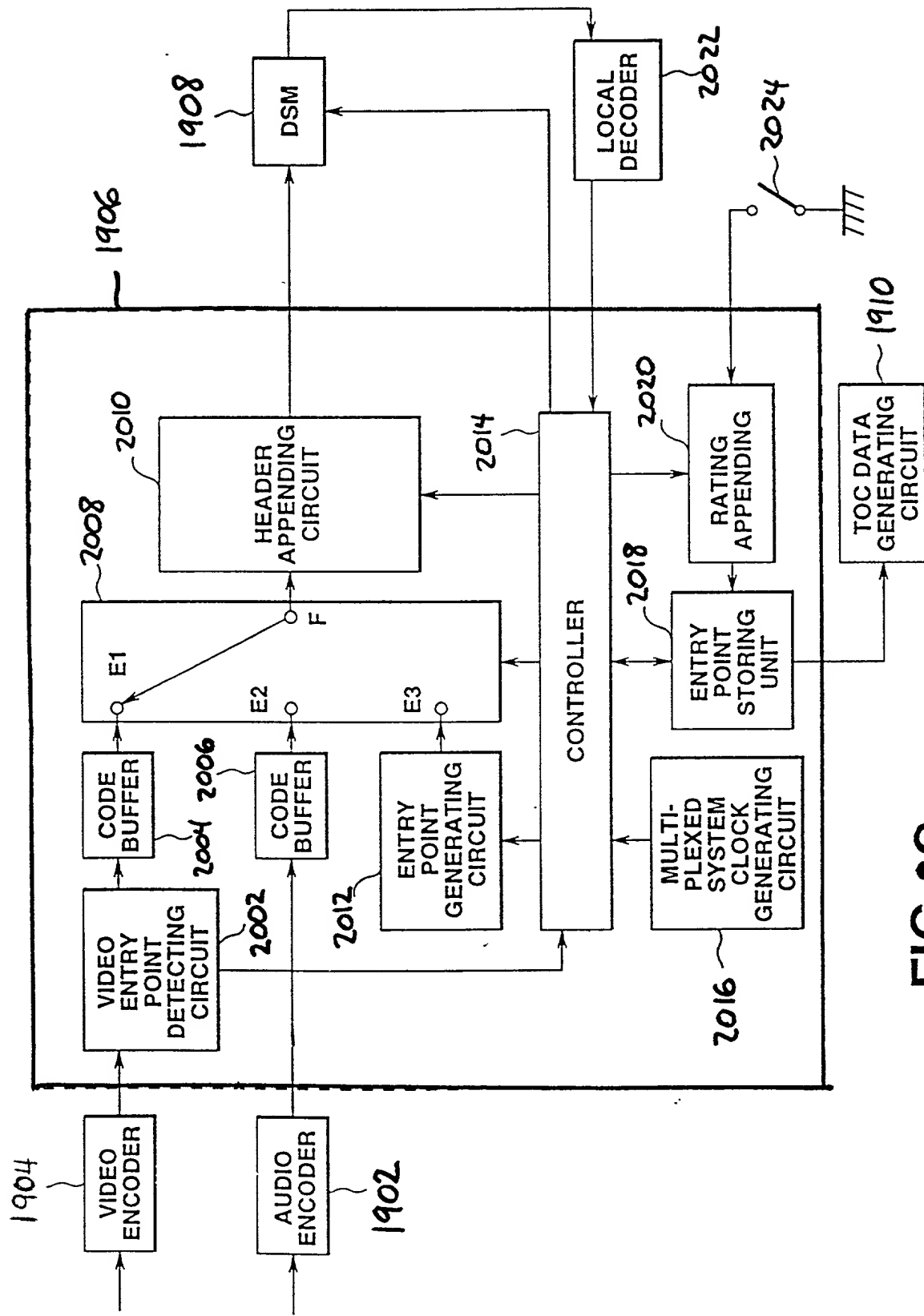


FIG.19



**FIG. 20**

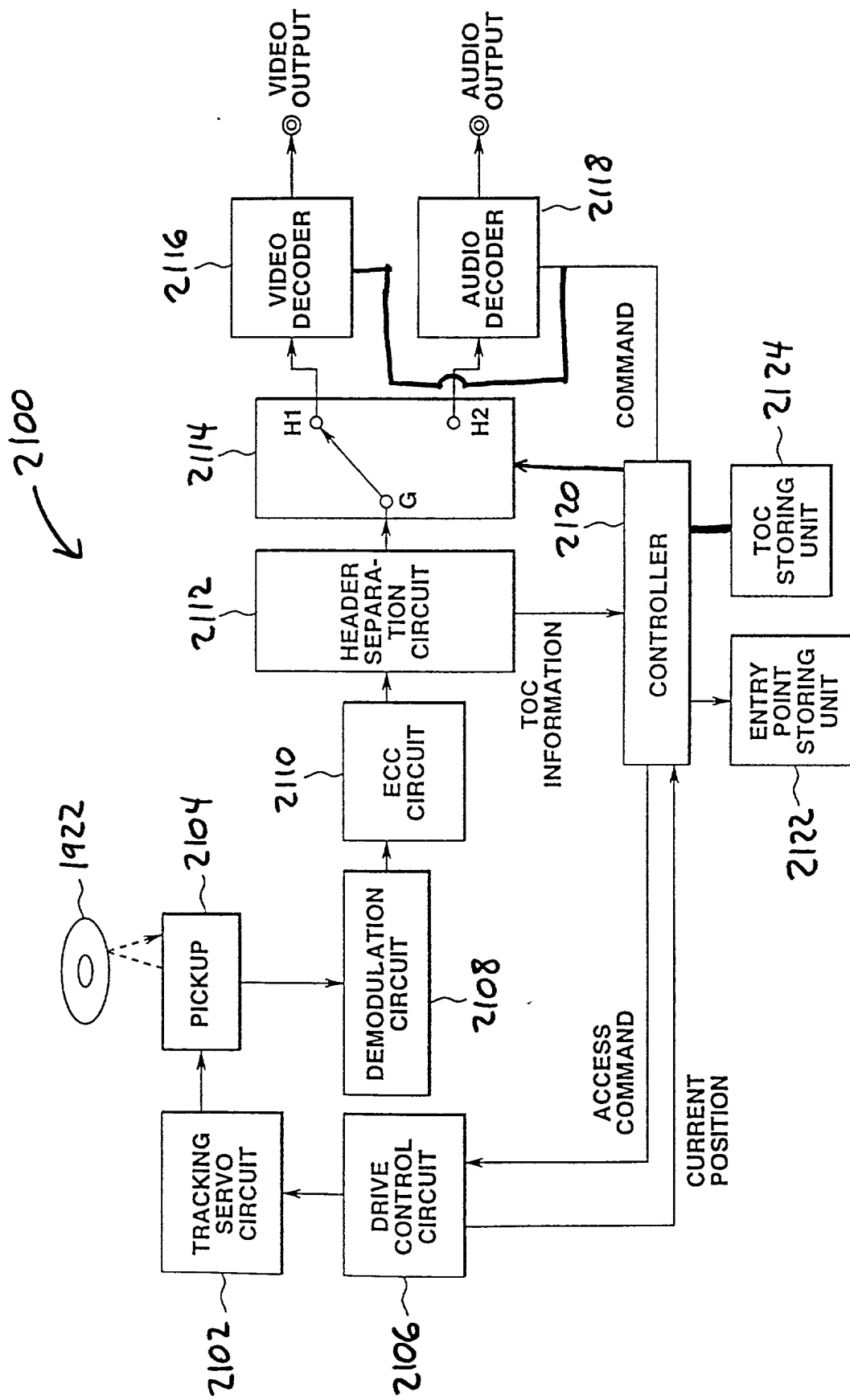


FIG. 21



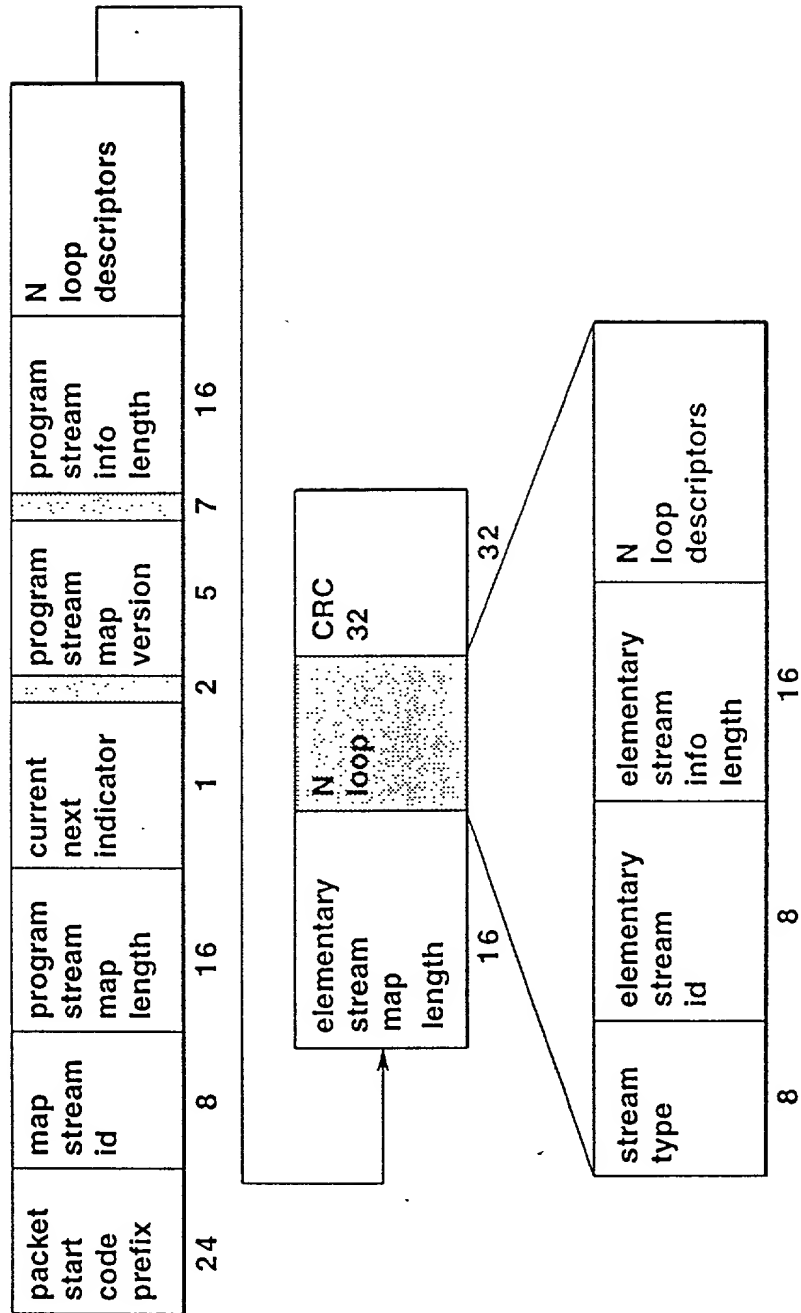
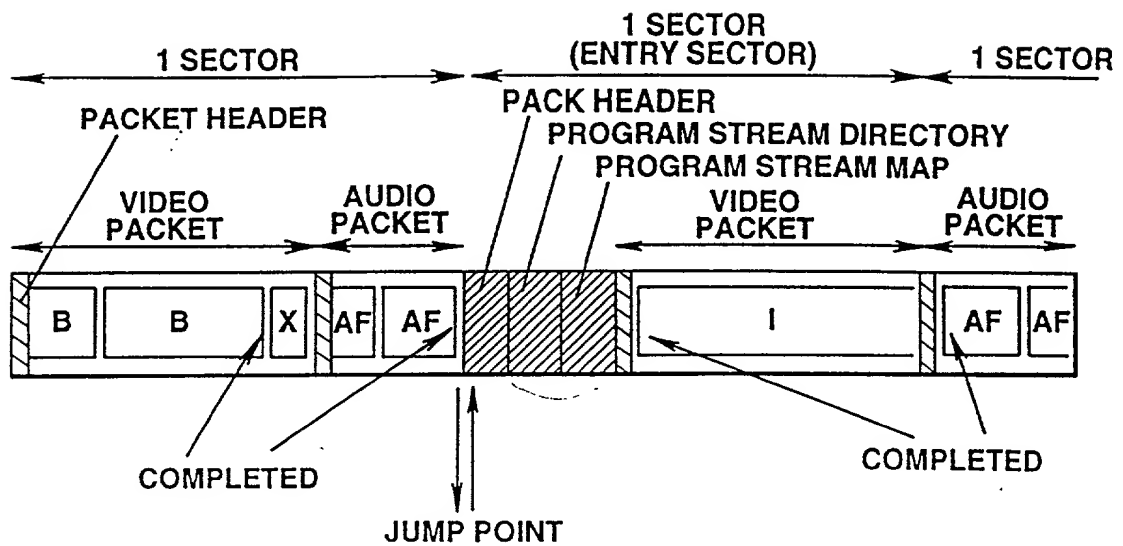


FIG. 22A

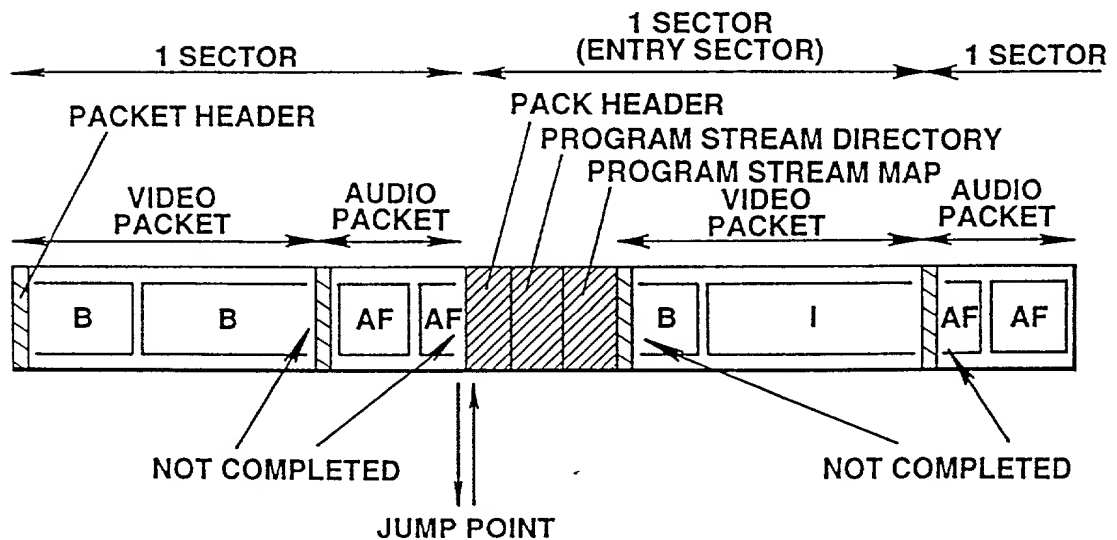
Syntax	No. of Bits	Mnemonic
program_stream_map(){		
packet_start_code_prefix	24	bslbf
map_stream_id	8	uimbsf
program_stream_map_length	16	uimbsf
current_next_indicator	1	bslbf
reserved	2	bslbf
program_stream_map_version	5	uimbsf
reserved	7	bslbf
marker_bit	1	bslbf
program_stream_info_length	16	uimbsf
for (i=0;i<N;i++){		
descriptor()		
}		
elementary_stream_map_length	16	uimbsf
for (i=0;i<N1;i++){		
stream_type	8	uimbsf
elementary_stream_id	8	uimbsf
elementary_stream_info_length	16	uimbsf
for (i=0;i<N2;i++){		
descriptor()		
}		
}		
CRC_32	32	rpchof
}		

FIG. 22B



EXAMPLE IN WHICH EACH ACCESS UNIT  
OF EACH ELEMENTARY STREAM IS  
COMPLETED AT JUMP POINT

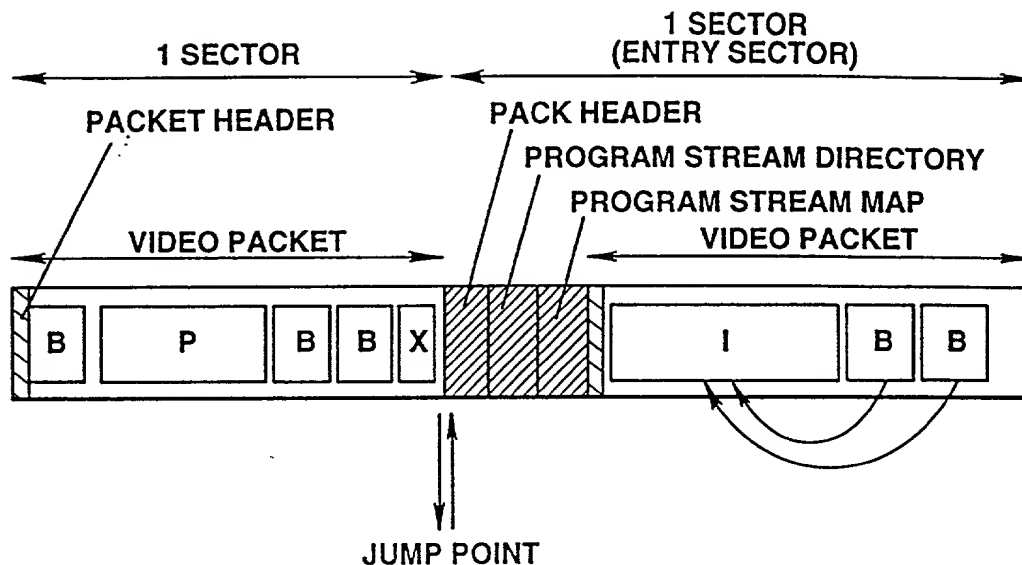
**FIG.23A**



X...PADDING DATA  
AF...AUDIO FRAME (AUDIO ACCESS UNIT)  
I ...I PICTURE (VIDEO ACCESS UNIT)  
B...B PICTURE (VIDEO ACCESS UNIT)

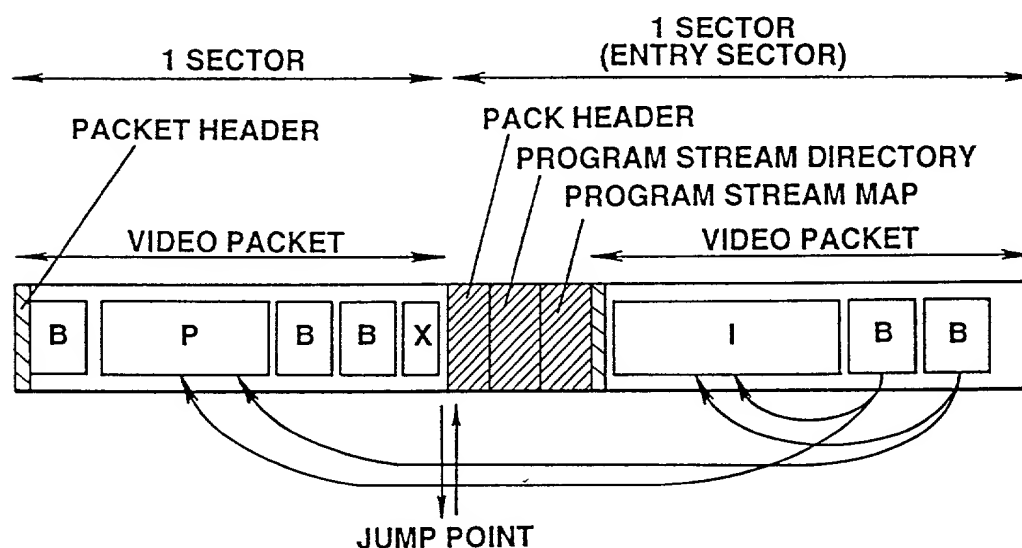
EXAMPLE IN WHICH EACH ACCESS UNIT  
OF EACH ELEMENTARY STREAM IS  
NOT COMPLETED AT JUMP POINT

**FIG.23B**



EXAMPLE IN WHICH PREDICTION  
(REFERENCE) IS NOT MADE ON  
BOTH SIDES OF JUMP POINT

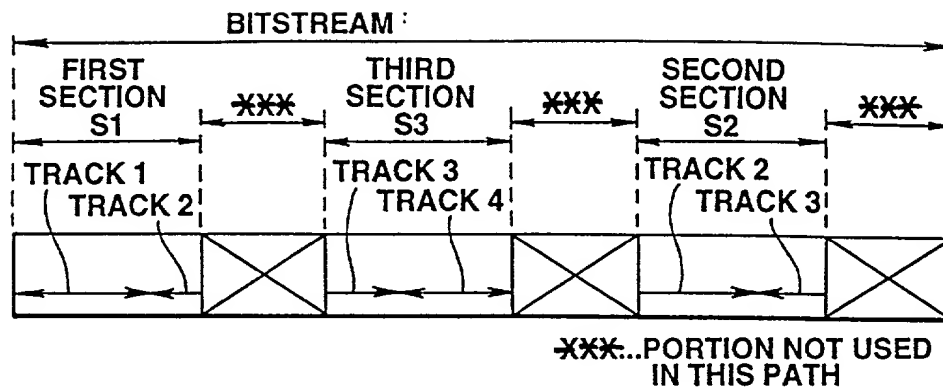
**FIG.24A**



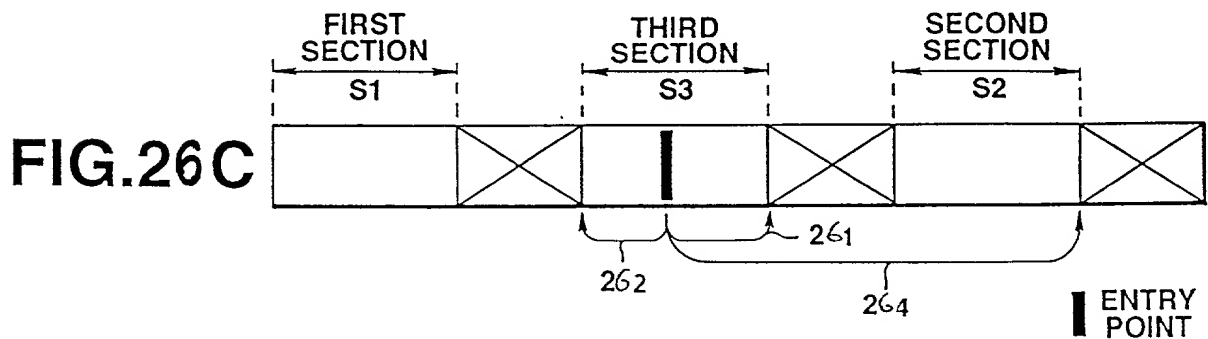
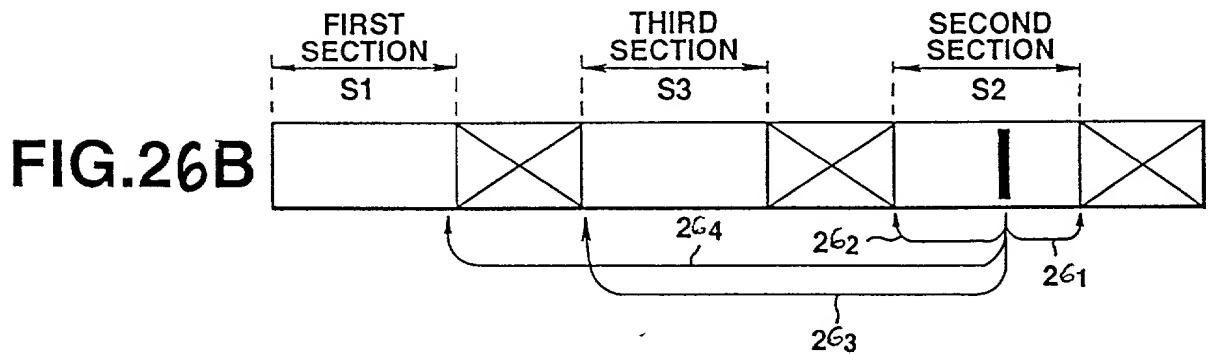
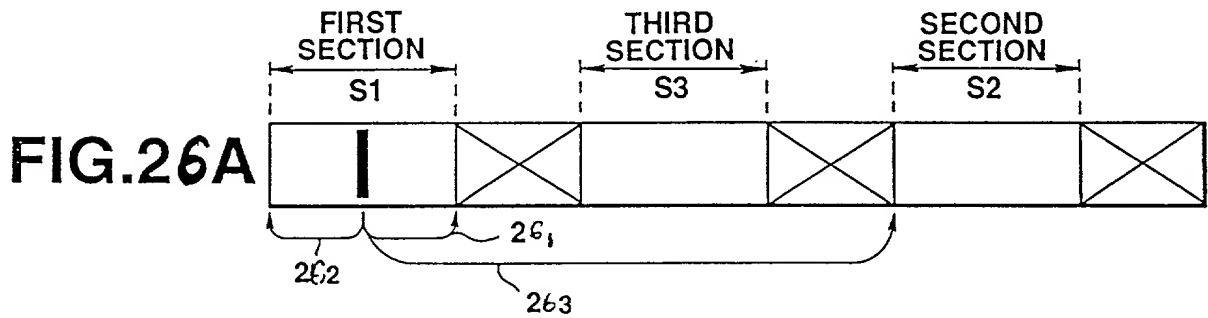
X...PADDING DATA  
AF...AUDIO FRAME (AUDIO ACCESS UNIT)  
I ...I PICTURE (VIDEO ACCESS UNIT)  
P...P PICTURE (VIDEO ACCESS UNIT)  
B...B PICTURE (VIDEO ACCESS UNIT)

EXAMPLE IN WHICH PREDICTION  
(REFERENCE) IS MADE ON  
BOTH SIDES OF JUMP POINT

**FIG.24B**

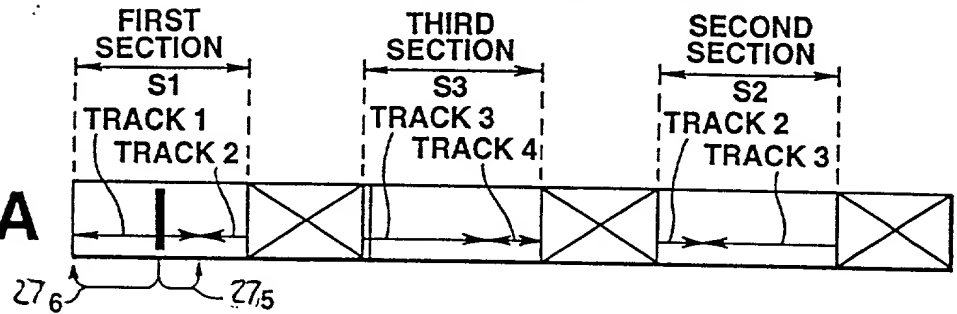


**FIG.25**



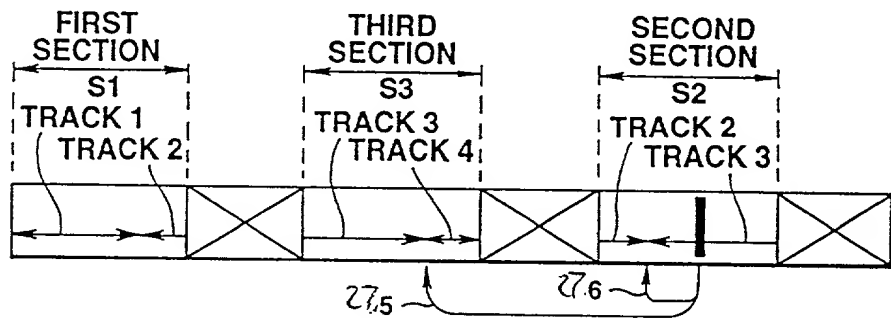
POSITION INDICATED BY TRACK POINTER OF  
AN ENTRY POINT IN FIRST SECTION

FIG.27A



POSITION INDICATED BY TRACK POINTER OF  
AN ENTRY POINT IN SECOND SECTION

FIG.27B



POSITION INDICATED BY TRACK POINTER OF  
AN ENTRY POINT IN THIRD SECTION

FIG.27C

